REMARKS

Claims 1-21 remain unchanged and are now pending in the application. The Examiner is respectfully requested to reconsider and withdraw the rejections in view of the amendments and remarks contained herein.

SPECIFICATION

The specification stands objected to for certain informalities. Applicants have amended the specification according to the Examiner's suggestions. Therefore, reconsideration and withdrawal of this objection are respectfully requested.

REJECTION UNDER 35 U.S.C. § 102

Claims 1-21 stand rejected under 35 U.S.C. § 102(a) as being anticipated by Convent (U.S. Pub. No. 2002/0016814). This rejection is respectfully traversed.

Independent Claim 1

Convent merely discloses that an application server 10 (see FIG. 1) generates an object 34 (see FIG. 2) based on result sets 62a to 62k and output parameters 64 (see FIG. 2) output from a stored procedure 28 in a database server 16, and transmits the object "itself" to a client system 4 (see FIG. 1). Therefore, Convent is different from the invention as recited in Claim 1 which transfers object "states".

Regarding the claimed limitation of "arranging the internal states of the plurality of objects into a byte sequence, which is manipulated from the application program via the accessor method", the Examiner cites the recitation of paragraph 0029, lines 9-15, of

Convent. However, the cited portion merely discloses that "for a class to be serializable, the user has to implement the java.io.Serializable interface and the class fields have to be either of a primitive type or serializable", and that "an object can be serializable if the class implements methods that write the state of non-primitive or non-serializable fields into the byte stream". As can also be understood from paragraph 0029, lines 6-9, of Convent, which recite that "Java Serialization is a standard Java mechanism that creates a platform independent byte stream of a Java objects state in order to allow the object to be written to a file or sent over a network", the cited portion merely relates to a general Java serialization. As explained in the "Description of the Related Art" of the specification of the present application, in order to perform serialization, the state of each individual member variable must be extracted by accessing each member variable, which represents the internal state of each instance, and the extracted states must be copied to a region for transfer where the byte sequence is to be stored. The cited portion merely discloses matters similar to those employed in the Related Art.

Unlike Convent, applicant's invention as recited in Claim 1 does not perform serialization. Rather, in the invention as recited in Claim 1, the internal states of a plurality of objects are arranged into a byte sequence and an application program manipulates the byte sequence via an accessor method. With such a feature, the states of the plurality of objects can be transferred to another data processing device by simply transmitting the byte sequence as it is to the other data processing device. Therefore, it is possible to avoid the overhead accompanying the serialization, which is a problem of the Related Art.

Moreover, to "write the state of non-primitive or non-serializable fields into the byte stream" recited in the cited portion is merely the writing of states into a byte sequence which is performed in a serialization process. This is different from the invention as recited in Claim 1 in which the application program manipulates "via the accessor" the byte sequence in which the internal states are arranged.

Regarding the claimed limitation of "transferring the internal states of the plurality of objects by transmitting the byte sequence to an external device", the Examiner points out "a request for data or other information from the client application" (paragraph 0029, lines 17-18, of Convent). However, such data or other information is merely data stored in a table 24 in a database 22 provided in the database server 16 (paragraph 0027, lines 14-15, of Convent). The data or other information of Convent is different from the claimed "byte sequence"; that is, the byte sequence in which "the internal states of the plurality of objected are arranged" and which is manipulated via the accessor method by the application program. A client application 6 merely returns an object "itself" (paragraph 0028, lines 6-8 of Convent). Moreover, the client application 6 merely returns a "single" object. This is different from the claimed byte sequence in which "the internal states" of "the plurality of" objects "are arranged".

Independent Claim 7

Claim 7 is a device claim which includes similar (common) limitations as those recited in method Claim 1. Therefore, the foregoing arguments based on the recitation of Claim 1 can apply to Claim 7.

In addition to the limitations similar to Claim 1, Claim 7 includes the limitation that stores mapping data relating to the internal states of the objects and the byte sequence and that transmits the byte sequence and the mapping data to another data processing device.

As shown in FIG. 2, Convent discloses input mappings 52 (paragraph 0029 pointed out by the Examiner), output mappings 60 (e.g., paragraph 0030), and an error mapping 80 (e.g., paragraph 0034). However, all of the three kinds of mappings are held in a remote interface implementation 12. Unlike the mapping data of Claim 7, the mappings of Convent are not transmitted to another data processing device, let alone transmitted to the other data processing device together with a byte sequence. Moreover, unlike the mapping data of Claim 7, the mappings of Convent are irrelevant to the internal states of "a plurality of objects" and a byte sequence.

Independent Claim 9

Claim 9 is an object state transfer device (the destination of transfer), which is a counterpart of an object state transfer device (the source of transfer) recited in Claim 7. Similar to Claim 7, Claim 9 includes the limitation that the internal states of a plurality of objects are arranged as a byte sequence and the limitation that mapping data is transferred. Therefore, regarding these limitations, the foregoing arguments based on the recitation of Claim 7 can apply to Claim 9.

In addition to the foregoing limitations, Claim 9 updates a byte sequence and mapping data within the object state transfer device itself based on byte sequence and mapping data received from another data processing device, reproduces objects having

the same state as the other data processing device based on the updated byte sequence and mapping data, and notifies an application program of data relating to the reproduced objects. These limitations of Claim 9 are neither disclosed nor suggested by the portions of Convent cited with respect to Claims 1 and 7.

Furthermore, in addition to the output mappings 60 set forth above, paragraph 0030 of Convent merely discloses that meta data 72 of the object 34 provides information on each of the elements added in the object 34, that is, which elements include output parameters, the data types and lengths of each output parameter, the number of different returned result sets, the structure of columns and data types in each result set, the number of rows, and how such result set data maps to the elements. Even if the meta data 72 of Convent were a kind of mapping, Convent merely generates the object 34 consisting of the meta data 72 and the respective elements 66 anew. Unlike Claim 9, Convent does not receive a byte sequence and mapping data from another data processing device, and update a byte sequence and mapping data within the object state transfer device itself based on the received byte sequence and mapping data.

Moreover, Convent receives data stored in the table 24 of the database 22 from the stored procedure 28 and generates the single object 34 anew. Moreover, since the output of the stored procedure 28 has a format which is inaccessible to the client application 6 (paragraph 0039, lines 14-16, of Convent), this output has a format different from that of the object 34 which is returned to the client application 6. Therefore, unlike Claim 9, Convent does not reproduce objects having the same state

as the other data processing device based on the updated byte sequence and mapping data, and notify an application program of data relating to the reproduced objects.

Therefore, it is respectfully submitted that Claims 1, 7 and 9, along with claims depending therefrom, define patentable subject matter over Convent. Accordingly, Applicant respectfully requests reconsideration and withdrawal of this rejection.

CONCLUSION

It is believed that all of the stated grounds of rejection have been properly traversed, accommodated, or rendered moot. Applicant therefore respectfully requests that the Examiner reconsider and withdraw all presently outstanding rejections. It is believed that a full and complete response has been made to the outstanding Office Action and the present application is in condition for allowance. Thus, prompt and favorable consideration of this amendment is respectfully requested. If the Examiner believes that personal communication will expedite prosecution of this application, the Examiner is invited to telephone the undersigned at (248) 641-1600.

Respectfully submitted,

Dated: October 8, 2008

By: /Timothy D. MacIntyre/_____
Timothy D. MacIntyre
Reg. No. 42,824

HARNESS, DICKEY & PIERCE, P.L.C.
P.O. Box 828
Bloomfield Hills, Michigan 48303
(248) 641-1600

TDM/dec